Homework 2

Deepika Dilip

3/20/2019

```
knitr::opts_chunk$set(echo=TRUE)
knitr::opts_chunk$set(warning=F)
knitr::opts_chunk$set(message=F)
```

```
library(caret) # only for plot
library(splines)
library(lasso2) # only for data
library(mgcv)
library(tidyverse)
library(ggplot2)
library(janitor)
library(boot) #for CV
```

Importing the Data

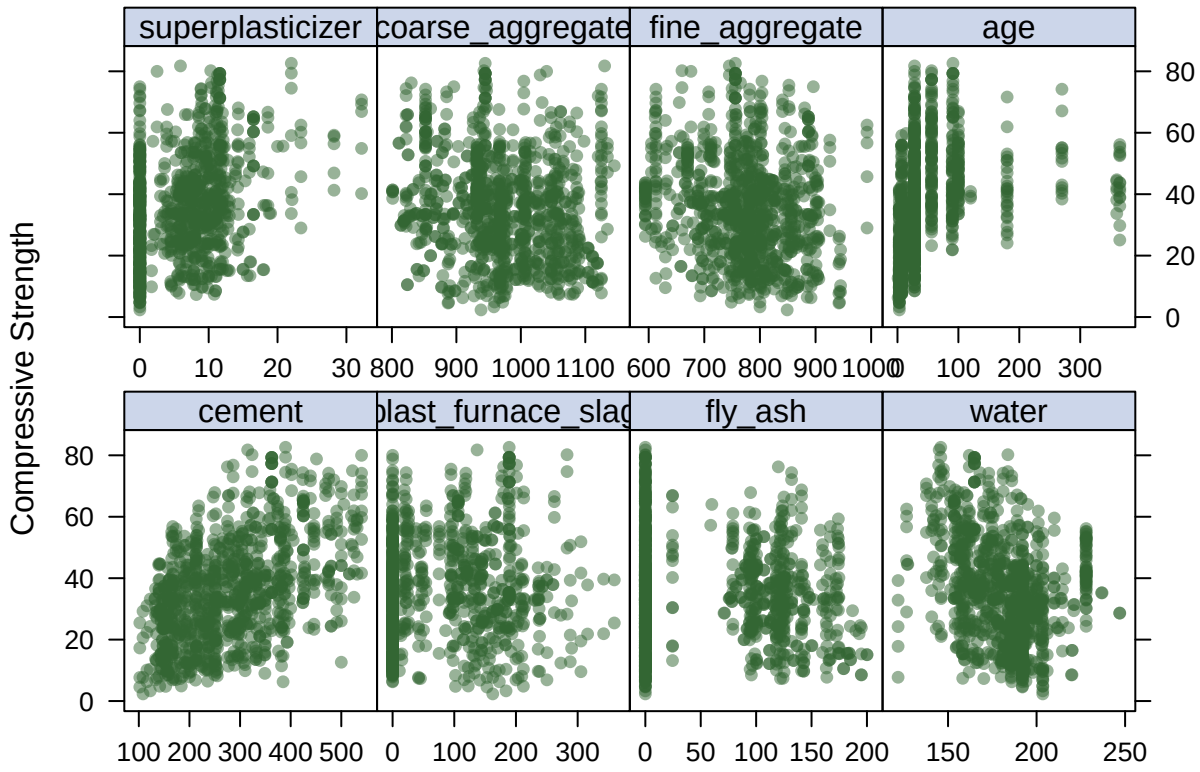
```
setwd('~/Dropbox/Spring 2019/Data Science II/Homework')
concrete <-read.csv("concrete.csv")
concrete= janitor::clean_names(concrete)
```

Defining the predictor and outcome matrices

```
# matrix of predictors
x <- model.matrix(compressive_strength~.,concrete)[,-1]
# vector of response
y <- concrete$compressive_strength
```

Part A: Plotting the Data

```
theme1 <- trellis.par.get()
theme1$plot.symbol$col <- rgb(.2, .4, .2, .5)
theme1$plot.symbol$pch <- 16
theme1$plot.line$col <- rgb(.8, .1, .1, 1)
theme1$plot.line$lwd <- 2
theme1$strip.background$col <- rgb(.0, .2, .6, .2)
trellis.par.set(theme1)
featurePlot(x, y, plot = "scatter", labels = c("","Compressive Strength"), type = c("p"), layout = c(4,
```



Part B: Polynomial regression

```
fit1 <- lm(compressive_strength~water, data = concrete)
fit2 <- lm(compressive_strength~poly(water,2), data = concrete)
fit3 <- lm(compressive_strength~poly(water,3), data = concrete)
fit4 <- lm(compressive_strength~poly(water,4), data = concrete)

cv.error=rep (0,5)
for (i in 1:5){
  glm.fit=glm(compressive_strength~poly(water,i), data=concrete)
  cv.error[i] = cv.glm(concrete, glm.fit)$delta[1]
}
print(cv.error)
```

```
## [1] 256.5072 242.3496 231.1924 226.6225 226.8308
```

```
anova(fit1,fit2,fit3,fit4)
```

```
## Analysis of Variance Table
##
## Model 1: compressive_strength ~ water
## Model 2: compressive_strength ~ poly(water, 2)
## Model 3: compressive_strength ~ poly(water, 3)
## Model 4: compressive_strength ~ poly(water, 4)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   1028 263085
## 2   1027 247712  1   15372.8 68.140 4.652e-16 ***
## 3   1026 235538  1   12174.0 53.962 4.166e-13 ***
```

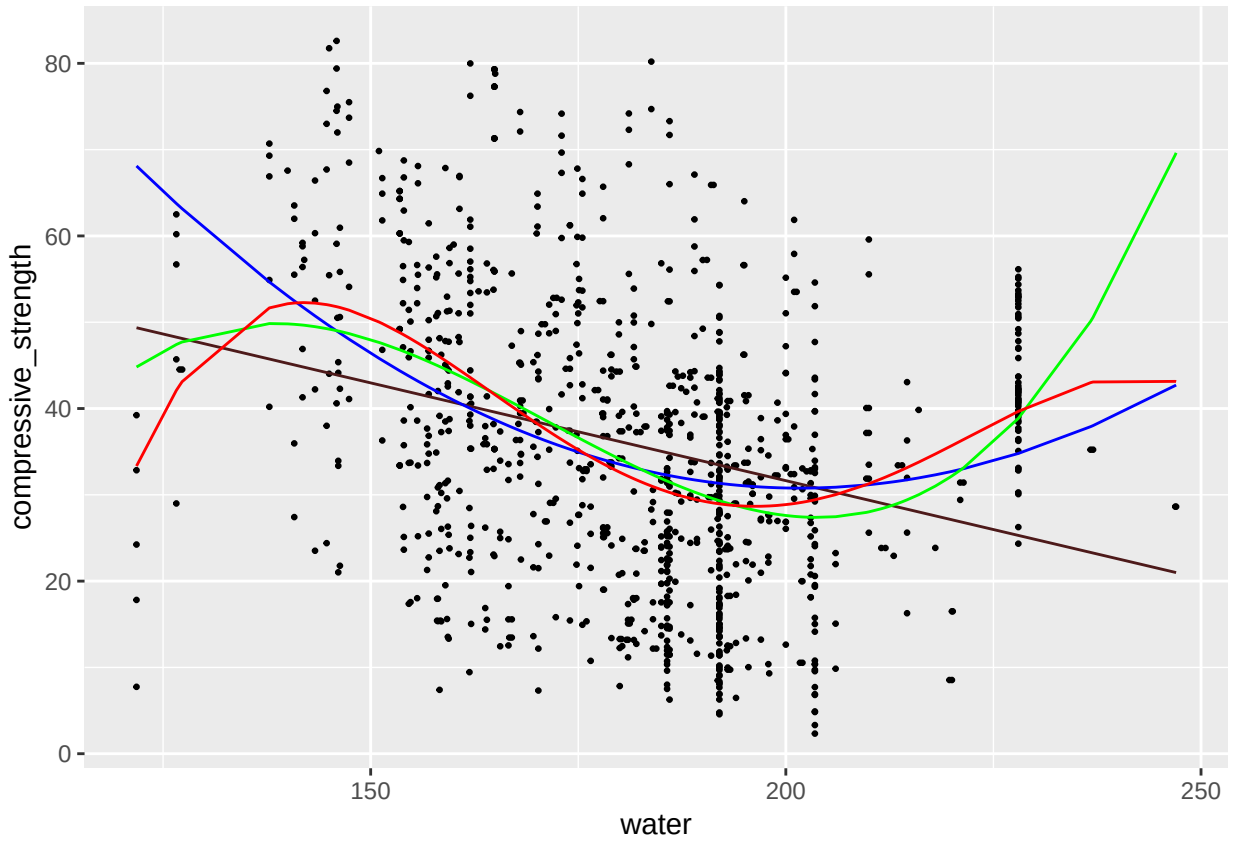
```
## 4   1025 231246  1    4291.5 19.022 1.423e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on cross validation, we want the d that minimizes the cross validation error. Therefore we should use d=4.

```
anova(fit1,fit2,fit3,fit4)
```

```
## Analysis of Variance Table
##
## Model 1: compressive_strength ~ water
## Model 2: compressive_strength ~ poly(water, 2)
## Model 3: compressive_strength ~ poly(water, 3)
## Model 4: compressive_strength ~ poly(water, 4)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   1028 263085
## 2   1027 247712  1   15372.8 68.140 4.652e-16 ***
## 3   1026 235538  1   12174.0 53.962 4.166e-13 ***
## 4   1025 231246  1    4291.5 19.022 1.423e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
pred.fit1 <- predict(fit1, x = water)
pred.fit2 <- predict(fit2, x = water)
pred.fit3 <- predict(fit3, x = water)
pred.fit4 <- predict(fit4, x = water)

p1 <- ggplot(data=concrete) +
  geom_point(aes(x=water, y=compressive_strength), color="black", size=0.5) +
  geom_line(aes(x = water, y = pred.fit1),
            color = rgb(.3, .1, .1, 1)) +
  geom_line(aes(x = water, y = pred.fit2),
            color = 'blue') +
  geom_line(aes(x = water, y = pred.fit3),
            color = 'green') +
  geom_line(aes(x = water, y = pred.fit4),
            color = 'red')
print(p1)
```

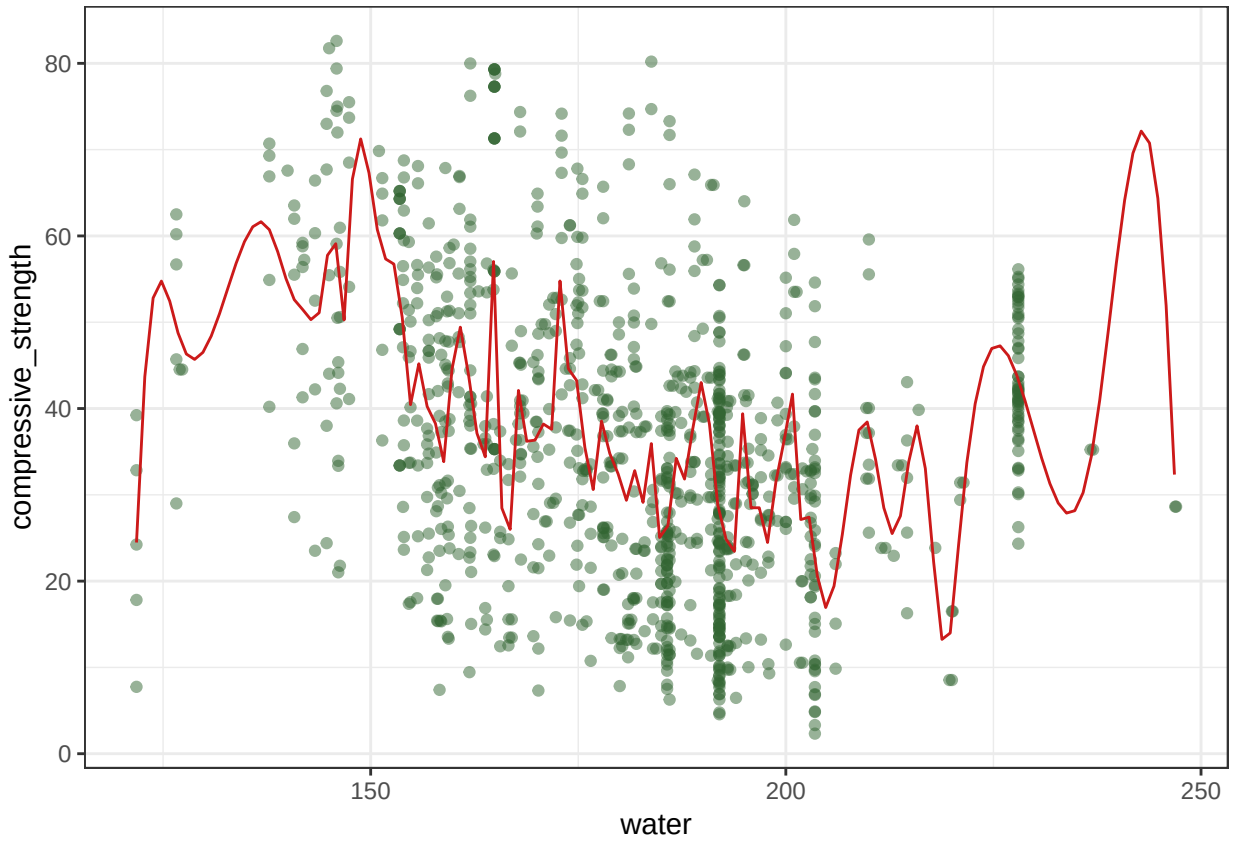


Part C: Smoothing spline

```
fit.ss <- smooth.spline(concrete$water, concrete$compressive_strength)
fit.ss$df
```

```
## [1] 68.88205
```

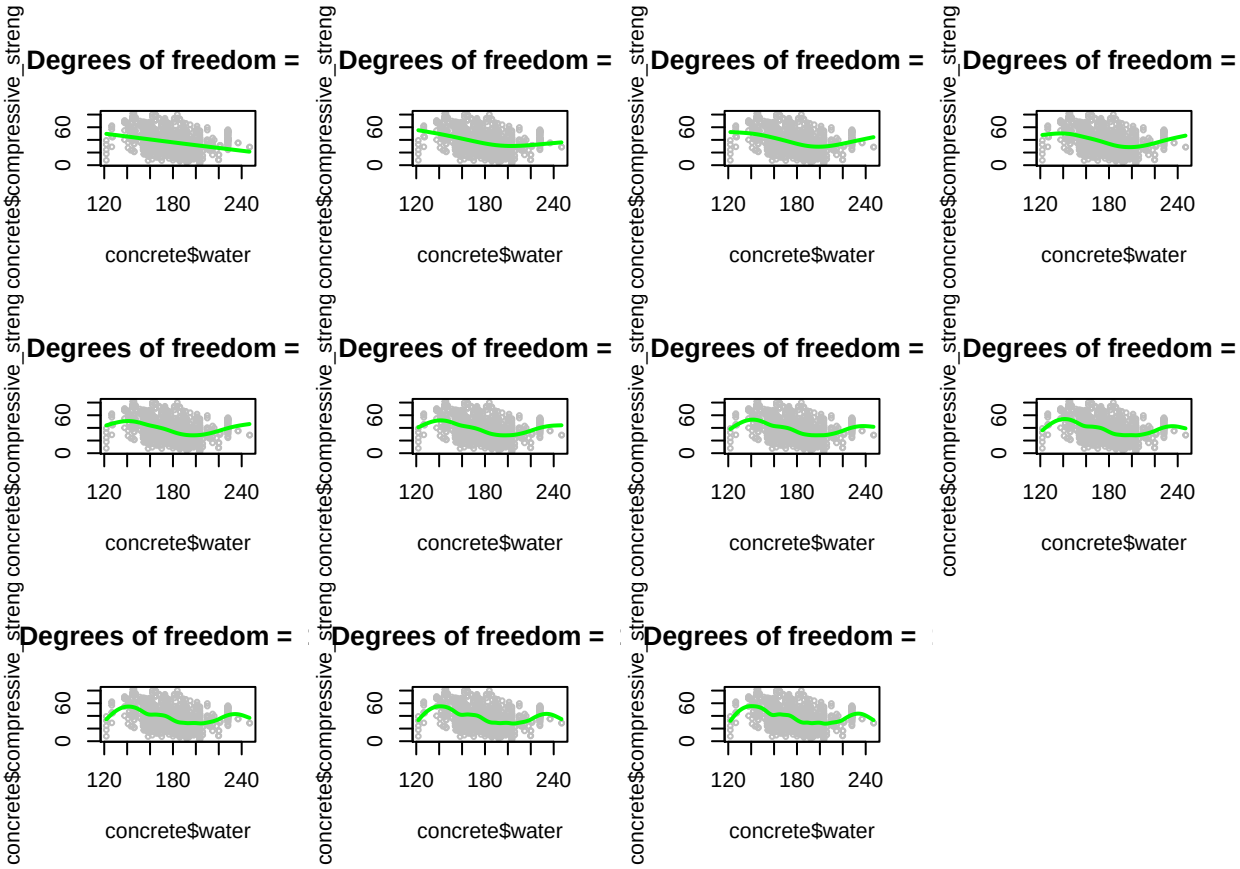
```
waterlimits <- range(concrete$water)
water.grid <- seq(from = waterlimits[1],to = waterlimits[2])

pred.ss <- predict(fit.ss, x = water.grid)
pred.ss.df <- data.frame(pred = pred.ss$y, water = water.grid)

spline_plot <- ggplot(data = concrete, aes(x = water, y = compressive_strength)) +
  geom_point(color = rgb(.2, .4, .2, .5)) +
  geom_line(aes(x = water, y = pred), data = pred.ss.df, color = rgb(.8, .1, .1, 1)) + theme_bw()

print(spline_plot)
```



##69 degrees of freedom

Range of df: 2-12

```
par(mfrow=c(3,4))
for (i in 2:12) {
  fit.ss1 = smooth.spline(concrete$water, concrete$compressive_strength, df = i)
  pred.ss_df <- predict(fit.ss1, x = water.grid)
  pred.ss_df <- data.frame(pred = pred.ss_df$y, water = water.grid)
plot(concrete$water, concrete$compressive_strength, cex = .5, col = "grey") + title(paste("Degrees of f
  }
```



Based on the results, it seems that df = 69 is most effective.

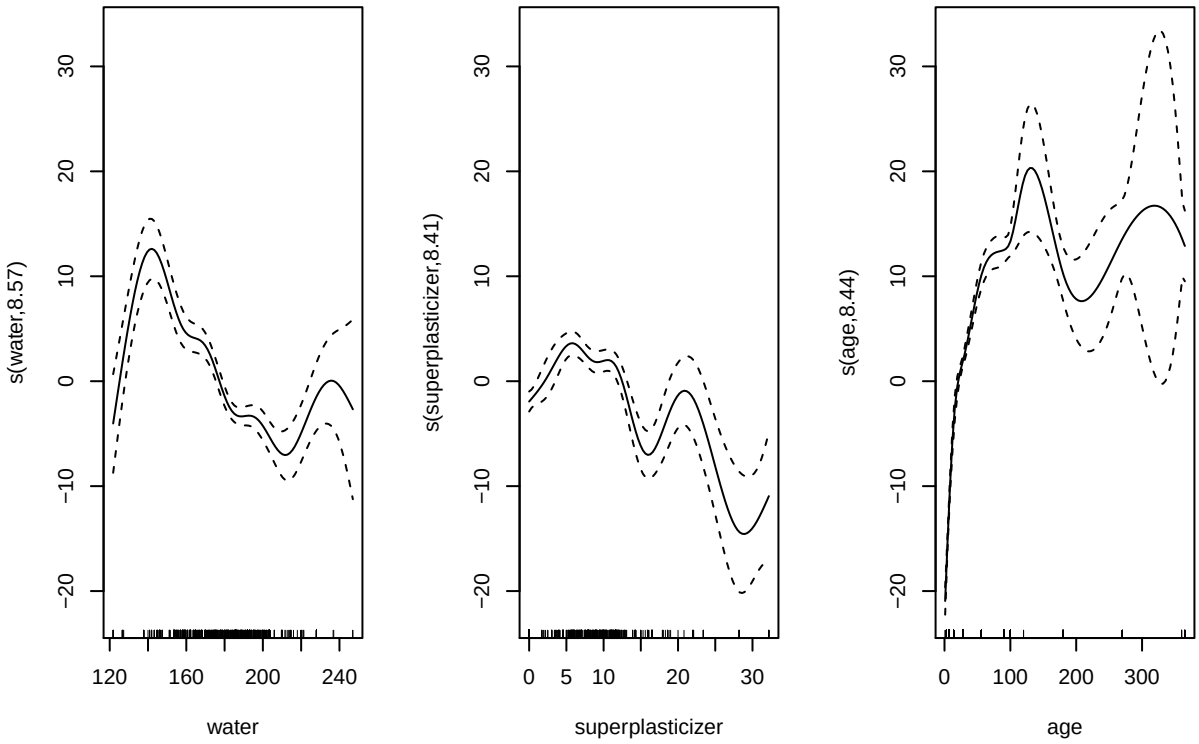
Part D: GAM

```
gam.m1 <- gam(compressive_strength ~ cement + blast_furnace_slag
              + fly_ash + water + superplasticizer + coarse_aggregate
              + fine_aggregate + age, data = concrete)
gam.m2 <- gam(compressive_strength ~ cement + blast_furnace_slag
              + fly_ash + s(water) + s(superplasticizer) + coarse_aggregate
              + fine_aggregate + s(age), data = concrete)
anova(gam.m1, gam.m2, test = "F")
```

```
## Analysis of Deviance Table
##
## Model 1: compressive_strength ~ cement + blast_furnace_slag + fly_ash +
##     water + superplasticizer + coarse_aggregate + fine_aggregate +
##     age
## Model 2: compressive_strength ~ cement + blast_furnace_slag + fly_ash +
##     s(water) + s(superplasticizer) + coarse_aggregate + fine_aggregate +
##     s(age)
##   Resid. Df Resid. Dev     Df Deviance      F    Pr(>F)
## 1   1021.00     110413
## 2    997.31      38093 23.691    72320 80.022 < 2.2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Plotting the Model
par(mfrow=c(1,3)) #to partition the Plotting Window
plot(gam.m2,se = TRUE)
```



Based on the scatter plots from Question 1, we can assume age, water, and superplasticizer require smoothing as they are nonlinear.